

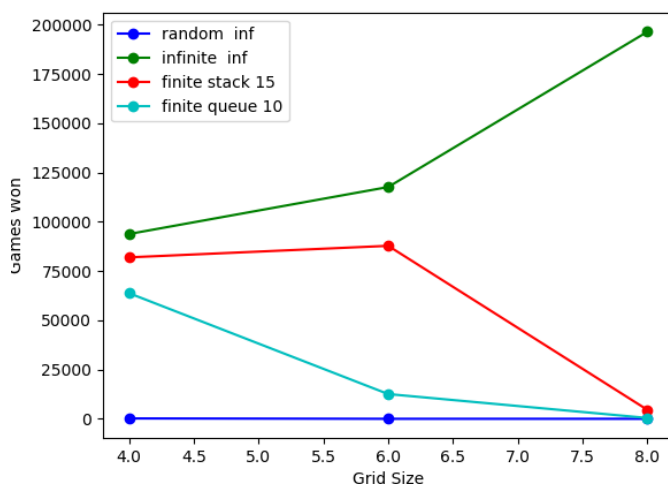
Správa IB111/01 – Miroslav Bezák - 485221

Projekt 1: Simulácia pexesa

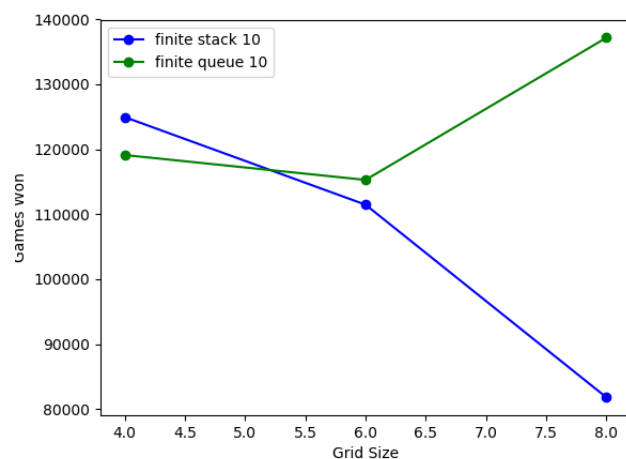
Úloha: Naprogramovať skript, ktorý vie simulovať ľubovoľný počet hier pexesa, na ľubovoľne veľkom hracom pláne a s voliteľným počtom hráčov. Ďalej skúmať rôzne druhy pamäti, ktorý môžu hráči používať a rozdiely medzi nimi. Pamäť môže byť: žiadna(random – celý čas náhodné vyberanie kartičiek), nekonečná(infinite) alebo konečná(možnosť si vybrať medzi ukladaním dátovou štruktúrou stack alebo queue a veľkosti tejto pamäti – koľko pozícií kartičiek si maximálne zapamätá).

Riešenie: Riadiaci class *PexesoGame* – pre každú simuláciu nový objekt, potom uloženie objektu s výsledkami hry a následná analýza celkových výsledkov. Class *Player*, ktorý si pamätá svoju stratégiu, podľa ktorej hrá(rôzne varianty metódy move), má vlastnú pamäť, ktorej limity mu určuje druh a veľkosť(*buffer_length* a *buffer_type*). Dekompozícia na 2 súbory – classes v jednom a skript na spustenie simulácií, analýzu výsledkov a vykreslenie grafov v druhom.

Výstupy: Grafy porovnávajúce súboje rôznych stratégií proti sebe



*všetky stratégie proti sebe – čím menší herný plán, tým vyrovnaneršie, stack ktorý mal väčšiu pamäť chvíľu dokázal konkurovať nekonečnej pamäti, ale na väčších plánoch mu obmedzená pamäť už chýbala.



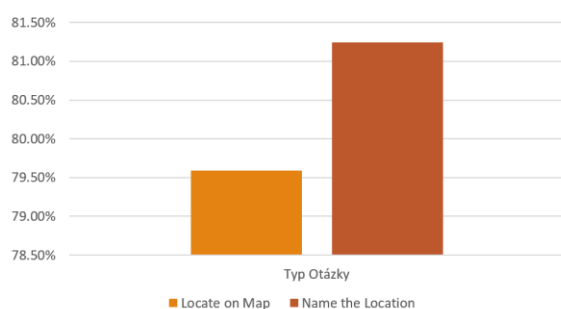
*porovnanie druhov pamätí – najprv vyrovnané, nakoniec sa queue ukázal ako výhodnejší – pamätá si aktuálnejšie kartičky

Projekt 2: Spracovanie dát

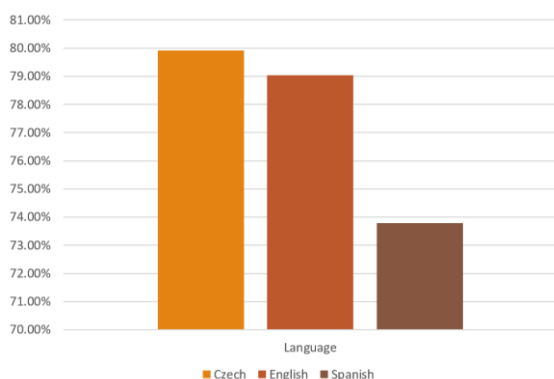
Úloha: Načítať a zanalyzovať dáta skupiny Adaptive Learning, konkrétne stránky slepemapy.cz. Snaha zodpovedať hlavnú otázku: Ktorý kontinent poznajú Česi a Slováci najlepšie? Pri spracovaní a skúmaní dát zodpovedať aj dielčie otázky: Je ľahšie pomenovať objekt alebo ho nájsť na mape? Akú majú používatelia úspešnosť vzhľadom na to, v akom jazyku stránku používajú? Ktorý typ geografického objektu vedie ľudia určiť najlepšie? V ktorú časť dňa je najvyššia úspešnosť?

Riešenie: Preloženie CSV súboru na slovník – kľúče sú názvy riadkov. Na dielčie otázky jednoduchšie štatistické výpočty. Hlavná otázka bola komplikovanejšia implementovať ako som pôvodne čakal, keďže súbory neobsahujú dáta, na akom kontinente sa miesta z otázok nachádzajú. Použil som preto API od Nokie (Mapy HereWeGo) – obmedzený počet requestov na mesiac – bol som nútený analyzovať iba menšiu vzorku dát.

Výstupy:

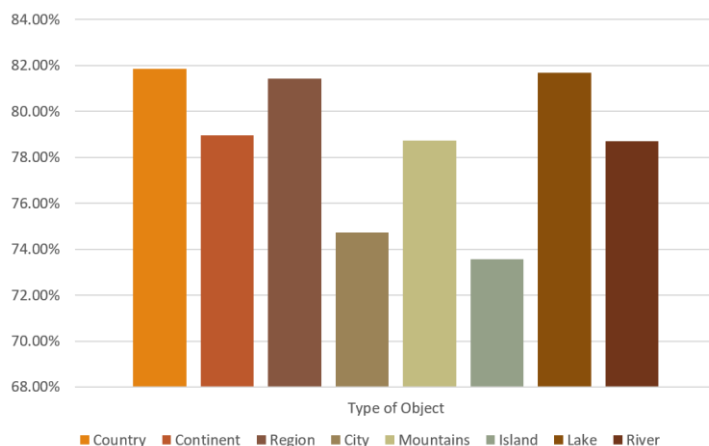


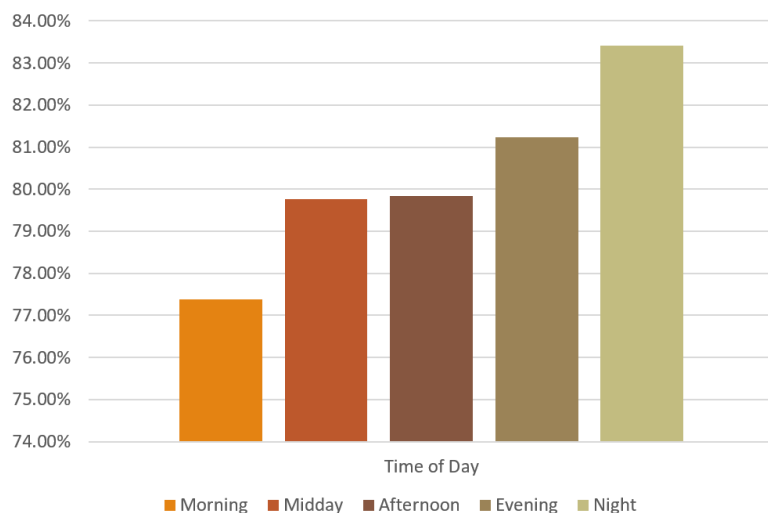
*je o trošku ľahšie pomenovať už vyznačené miesto, ako hľadať jeho polohu na slepej mape



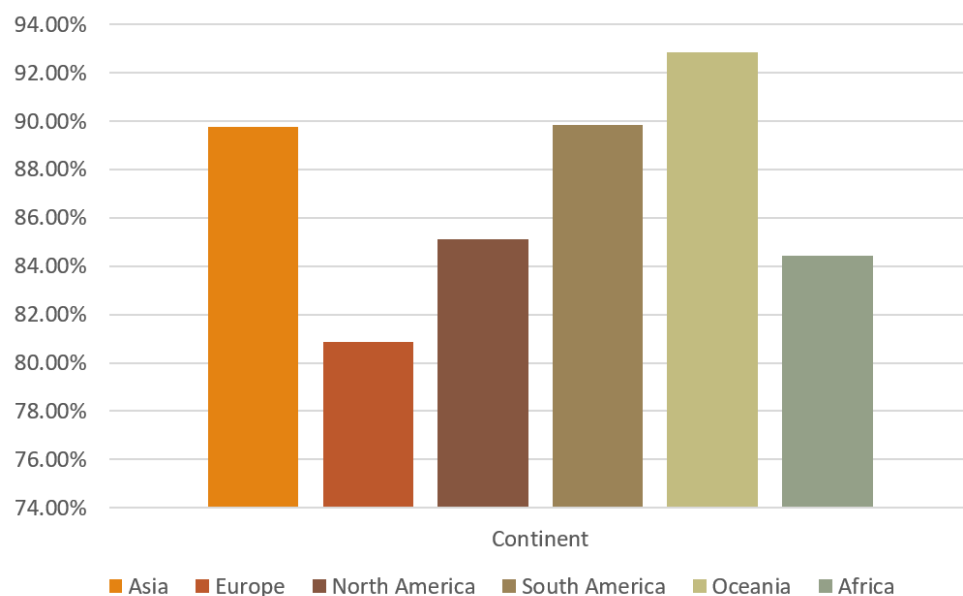
*najvyššiu úspešnosť mali ľudia s češtinou – kvôli tomu, že ich bola väčšina a otázky na Česko a Slovensko boli najnáročnejšie

*najvyššia úspešnosť krajina, región – objekty s najväčšou plochou, najmenej triviálne otázky





*najväčšia úspešnosť večer a v noci
– najmenej používateľov, a zároveň
tí starší alebo ľudia venujúci sa
geografii – najzdatnejší, ráno
znižujú úspešnosť napríklad mladší
žiaci, ktorí sa geografii ešte len učia



*výsledky hlavnej analýzy neboli také, ako som pôvodne očakával – viacero dôvodov: malá vzorka dát, obmedzené kvôli limitu externej API a tiež kvôli zloženiu otázok na slepemapy.cz, otázky boli zamerané na nám najbližšie kontinenty (Európu a z časti Severnú Ameriku), na vedomosť zvyšných kontinentov nebol kladený taký dôraz a boli na iba jednoduché otázky, ktoré väčšina používateľov zvládla.

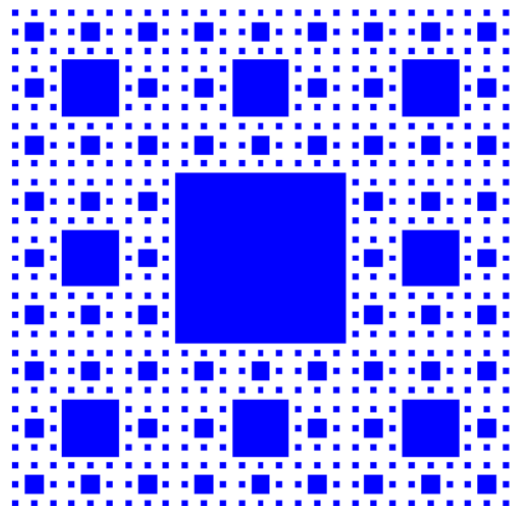
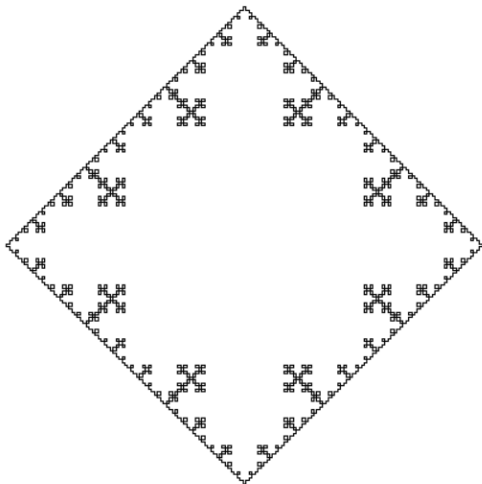
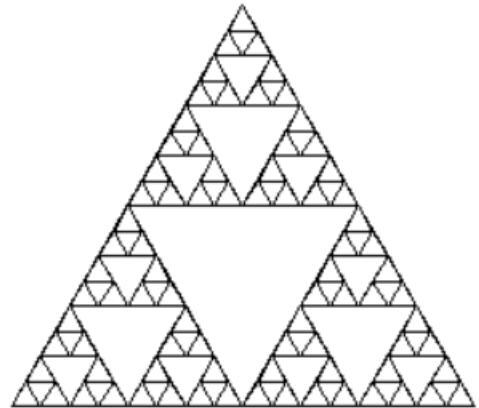
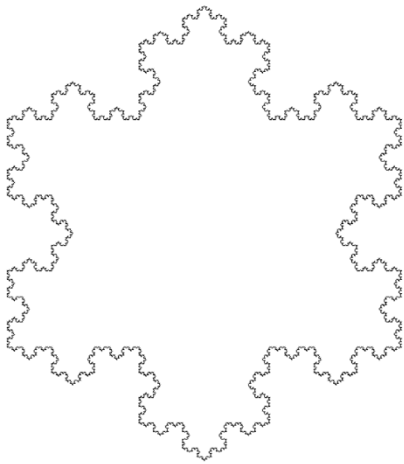
(.00 z percent v grafoch nešlo odstrániť, pretože grafy boli generované z textového výstupu, ktorý je na 2 desatinné miesta a stupnica sa generuje automaticky)

Projekt 3a: Rekuzívne Fraktály

Úloha: Pomocou knižnice Turtle implementovať niekoľko klasických fraktálov a následne aj vlastných variácií. Ja som si vybral Sierpinskeho trojuholník, Kochovu vložku a úpravy: Kochov štvoruholník(vložka založená na štyroch hranách miesto troch) a Sierpinskeho koberec.

Riešenie: Klasická rekurzia, na trojuholník použité hľadanie stredu úsečky a geometricky vypočítané súradnice v *base_triangle* aby bol trojuholník rovnostranný. Na vložku a štvoruholník použité otáčanie korytnačky o vnútorný resp. vonkajší uhol a následné vnorenie sa pre každú čiaru. Pri koberci som od prezentácie trochu zmenil postup, pretože ten pôvodný trval hrozne dlho vykresliť, zoberieme si stredný štvorec a potom pridáme 8 okolo a na každý z nich zase pridáme osem rekuzívne až do limitu vnorenia.

Výstupy:

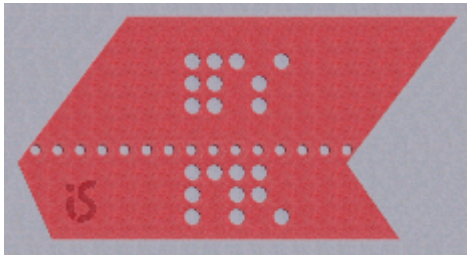


Projekt 3b: Spracovanie diernej pásky

Úloha: Napísať skript, ktorý vyčíta z obrázkov diernej pásky (Laborať predmetu PB151 Výpočetní systémy) jednotlivé ASCII znaky a vypíše ich.

Riešenie: Za použitia rôznych funkcionalít knižnice Image je po častiach problém relatívne jednoducho vyriešiť. Pracujeme s 2D listom pixelov. Aplikujeme bilivel filter z Image, ktorý nám obrázok prekonvertuje na Black&White, následne definujeme dierku ako sekvenciu 8ich bielych pixelov a podľa ich pozície určíme ktoré mocniny dvojky boli v ktorom stĺpci vydierkované.

Výstupy: Bolo testované iba na priložených vstupoch vzhľadom na neverejnosť generátora týchto obrázkov.



*znaky ASCII: wtG6A